

This Page Is Inserted by IFW Operations
and is not a part of the Official Record

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

IMAGES ARE BEST AVAILABLE COPY.

**As rescanning documents *will not* correct images,
please do not report the images to the
Image Problem Mailbox.**

(19)



JAPANESE PATENT OFFICE

JPA2002-328757

PATENT ABSTRACTS OF JAPAN

(11) Publication number: 2002328757 A

(43) Date of publication of application: 15.11.02

(51) Int. Cl

G06F 3/00
G06F 3/12

(21) Application number: 2001132928

(71) Applicant: CANON INC

(22) Date of filing: 27.04.01

(72) Inventor: MINAGAWA TOMONORI

(54) USER INTERFACE CONTROLLER AND USER
INTERFACE CONTROL METHOD

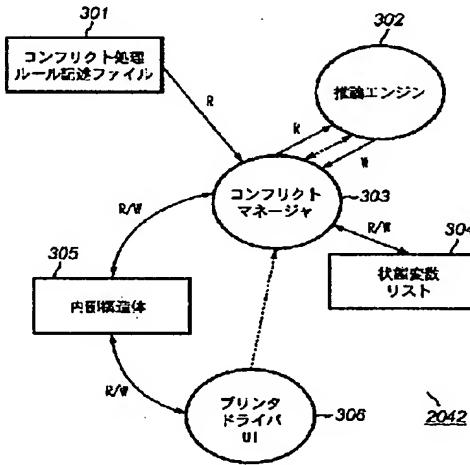
applies this rule.

COPYRIGHT: (C)2003,JPO

(57) Abstract:

PROBLEM TO BE SOLVED: To provide a user interface controller and a user interface control method capable of reducing the work man-hours or artificial mistake of a developer by realizing a function for displaying a reason why the setting of a specific item is made impossible without occurring the omission of the detection of the reason.

SOLUTION: A rule for preventing any mismatching generated between set data is described in a conflict processing rule description file (301). An instruction to make it impossible to directly change the state of a prescribed set item after applying the specific conflict processing rule and an identifier indicating a reason why the direct change is made impossible according to the instruction can be described in the conflict processing rule description file (301). An inference engine (302) refers to the conflict processing rule description file (301), and when the input set value is turned into an object to which any conflict processing rule should be applied, the inference engine (302)



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2002-328757

(P2002-328757A)

(43)公開日 平成14年11月15日 (2002.11.15)

(51) Int.Cl.⁷
G 0 6 F 3/00
3/12識別記号
6 5 4F I
G 0 6 F 3/00
3/12テーマコード(参考)
6 5 4 A 5 B 0 2 1
C 5 E 5 0 1

審査請求 未請求 請求項の数9 O L (全 13 頁)

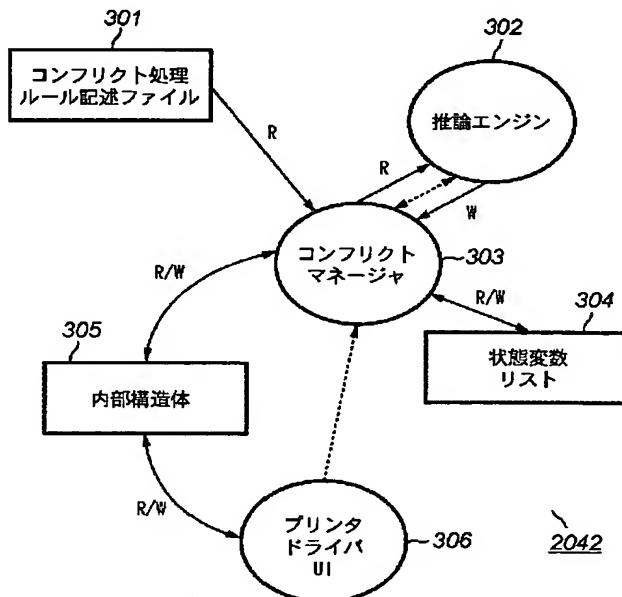
(21)出願番号 特願2001-132928(P2001-132928)
(22)出願日 平成13年4月27日(2001.4.27)(71)出願人 000001007
キヤノン株式会社
東京都大田区下丸子3丁目30番2号
(72)発明者 皆川 智徳
東京都大田区下丸子3丁目30番2号 キヤ
ノン株式会社内
(74)代理人 100076428
弁理士 大塚 康徳 (外3名)
Fターム(参考) 5B021 AA01 BB01 CC05 PP04
5E501 AA01 AA06 BA05 CA03 CB02
CB09 EA10 EA11 FA03 FA04
FA13 FA23 FA43 FA45 FB43

(54)【発明の名称】 ユーザインターフェース制御装置および方法

(57)【要約】

【課題】 特定の項目が設定不可となった理由を表示する機能を、理由検知の漏れ等を起こすことなく実現し、開発者の作業工数や人為的ミスの低減が可能なユーザインターフェース制御装置および方法を提供すること。

【解決手段】 設定データ間に生じる不整合を回避するためのルールがコンフリクト処理ルール記述ファイル(301)に記述される。このコンフリクト処理ルール記述ファイル(301)には、特定のコンフリクト処理ルールの適用後、所定の設定項目の状態を直接変更不可とするよう指示するための命令と、その命令によって前記直接変更不可とされた理由を示す識別子をあわせて記述することを可能にする。推論エンジン(302)はこのコンフリクト処理ルール記述ファイル(301)を参照し、入力設定値がいずれかのコンフリクト処理ルールの適用対象となったときに当該ルールを適用する。



【特許請求の範囲】

【請求項 1】 ユーザインタフェースを介して入力される所定の被制御装置に対する設定情報および／または当該被制御装置の環境情報の間に生じる不整合を回避するためのユーザインタフェース制御装置であって、前記設定情報および／または前記環境情報に応じて適用される不整合回避戦略を示すコンフリクト処理ルールを保持する保持手段と、前記コンフリクト処理ルールに従って、前記ユーザインタフェースにおける所定の設定項目の状態の直接変更の可否を制御する状態制御手段と、該状態制御手段により前記所定の設定項目の状態が直接変更不可とされた理由を含むメッセージを前記ユーザインタフェースを介して通知する通知制御手段と、を備え、

前記コンフリクト処理ルールは、所定の設定項目の状態を直接変更不可とするための命令と、該命令によって前記直接変更不可とした理由を示す理由情報と、を含むことを特徴とするユーザインタフェース制御装置。

【請求項 2】 前記状態制御手段は、前記設定情報および／または前記環境情報に応じて適用される前記コンフリクト処理ルールに含まれる前記命令に従って、前記ユーザインタフェースにおける所定の設定項目の状態の直接変更の可否を制御することを特徴とする請求項 1 に記載のユーザインタフェース制御装置。

【請求項 3】 前記通知制御手段は、前記適用されたコンフリクト処理ルールに含まれる前記理由情報に応じた前記メッセージを前記ユーザインタフェースを介して通知することを特徴とする請求項 1 または 2 に記載のユーザインタフェース制御装置。

【請求項 4】 前記コンフリクト処理ルールはコンフリクト処理ルール記述ファイルに記述されて記憶され、該コンフリクト処理ルール記述ファイルにおける各コンフリクト処理ルールは所定の優先順位を有し、前記状態制御手段が、一の設定項目に対して複数の前記コンフリクト処理ルールを重複適用した場合において、前記通知制御手段は、前記重複適用されたコンフリクト処理ルールのうち前記優先順位が最も高いコンフリクト処理ルールに含まれる前記理由情報に応じたメッセージを前記ユーザインタフェースを介して通知することを特徴とする請求項 3 に記載のユーザインタフェース制御装置。

【請求項 5】 前記優先順位は、前記コンフリクト処理ルール記述ファイルにおけるコンフリクト処理ルールの記述順に従って定められることを特徴とする請求項 4 に記載のユーザインタフェース制御装置。

【請求項 6】 前記制御対象は、画像形成装置であるこ

とを特徴とする請求項 1 ないし 5 のいずれか 1 項に記載のユーザインタフェース制御装置。

【請求項 7】 ユーザインタフェースを介して入力される所定の制御対象に対する設定情報および／または当該制御対象の環境情報の間に生じる不整合を回避するためのユーザインタフェース制御方法であって、

前記設定情報および／または前記環境情報に応じて適用される不整合回避戦略を示すコンフリクト処理ルール

(当該コンフリクト処理ルールは、所定の設定項目の状態を直接変更不可とするための命令と、該命令によって前記直接変更不可とした理由を示す理由情報とを含む) に従って、前記ユーザインタフェースにおける所定の設定項目の状態の直接変更の可否を制御する状態制御工程と、

前記適用されたコンフリクト処理ルールに含まれる前記理由情報に応じたメッセージを、前記ユーザインタフェースを介して通知する通知制御工程と、

を有することを特徴とするユーザインタフェース制御方法。

20 【請求項 8】 コンピュータ装置が実行可能なプログラムであって、当該プログラムを実行したコンピュータ装置を、請求項 1 ないし 6 のいずれか 1 項に記載のユーザインタフェース制御装置として機能させることを特徴とするプログラム。

【請求項 9】 請求項 8 に記載のプログラムを格納した記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、ユーザインタフェースを介して入力される所定の制御対象に対する設定データの間に生じる不整合を回避するユーザインタフェース制御装置および方法に関する。

【0002】

【従来の技術】 ユーザインタフェース（以下、「UI」ともいう。）を介してユーザから複数の設定値の入力を受け付け、それらの設定値に基づき制御される機器は、数多くある。ホストコンピュータに接続され、そのホストコンピュータにおいて設定された情報に基づき画像形成処理を行う画像形成システム（プリンタシステム）は

40 その一例である。プリンタシステムのホストコンピュータは一般に、印刷動作を制御するいわゆるプリンタドライバ、およびユーザからの印刷設定等を受け付ける UI を含む印刷処理関連プログラムを有する。

【0003】 印刷処理関連プログラムは、UI を介してユーザから設定値の入力を受け付ける度に、それまでに設定された複数の設定値の中で関連する設定値の値との関係を評価し、設定値間の不整合（コンフリクト）がないかどうかを判別する。コンフリクトの例としては、記録媒体としてセットされたOHPシートに対して両面印刷を行わせるような設定等のユーザにとって不都合と予

50 刷を行わせるような設定等のユーザにとって不都合と予

想される設定や、プリンタに不可能な動作を行わせるような設定等がある。

【0004】コンフリクトがあった場合には、それを回避するためのコンフリクト処理を実施する必要がある。

【0005】従来より、コンフリクト処理が必要となる複数の設定値の条件（コンフリクト処理ルール）を一覧としてまとめてファイル等に保存しておき、このファイルをコンフリクト処理用プログラムに読み込ませることで、コンフリクト処理を行うものがある。かかる方式は、コンフリクト処理用プログラムが特定の設定値に依存することなく汎用的に利用できるという点で優れている。

【0006】ところで、コンフリクト処理によっては、データ構造体中の設定値の変更のみならず、UIのコントロール（以下、単に「コントロール」ともいう。）上の設定値の変更、コントロールをグレイアウト（設定項目を薄いグレイで表示すること）して当該項目は（マウス等によって）直接変更することができないことをユーザに知らしめる処理等を行うものもある。

【0007】このような処理を含むコンフリクト処理により、他の項目に依存して利用できなくなったコントロールは、上記したようにグレイアウトされることで直接変更できないことが分かるようになっているものの、直接変更できなくなった原因（すなわち、依存関係のある他の項目）が、複数のシートやダイアログボックスのうち別のシートまたはダイアログボックスにあるような場合には、ユーザにはその原因が分かりにくい。この問題を解消するために、グレイアウトされたコントロールの付近に、当該コントロールが直接変更できない理由を含むメッセージを表示するものもある。

【0008】

【発明が解決しようとする課題】ところで、従来、上記したような、コントロールがグレイアウトされて直接変更できない理由を表示するためには、コンフリクト処理の記述とは別に、理由検知専用のプログラムを作成する必要があった。理由検知を正確に行うためには、グレイアウトするための処理の各々に対応する理由検知処理のプログラムコードを必要とする。そして、このプログラムコードの作成には手間がかかるだけでなく、グレイアウト処理の追加／変更に伴って理由検知処理の漏れや食い違いが出やすいという問題があった。

【0009】また、例えば、特定のデバイスオプション（例えばステープル用のフィニッシャ）が装着されていないと当該コントロールが使用できず、なおかつ、他の項目とも排他関係にある等、当該コントロールがグレイアウトされる理由が複数存在する場合もある。そのような場合には、理由検知処理がさらに複雑になり、プログラムの作成に関する上記の問題はより顕著なものとなる。

【0010】本発明はこのような問題点に鑑みてなされ

たもので、ユーザインタフェースにおいて特定の項目が設定不可となった理由を表示する機能を、理由検知の漏れや食い違いを起こすことなく実現し、開発者の作業工数や人為的ミスを低減することの可能なユーザインタフェース制御装置および方法を提供することを目的とする。

【0011】

【課題を解決するための手段】上記目的を達成するため、例えば本発明のユーザインタフェース制御装置は、

- 10 以下の構成を備える。すなわち、ユーザインタフェースを介して入力される所定の制御対象に対する設定情報および／または当該制御対象の環境情報の間に生じる不整合を回避するためのユーザインタフェース制御装置であって、前記設定情報および／または前記環境情報に応じて適用される不整合回避戦略を示すコンフリクト処理ルールを保持する保持手段と、前記コンフリクト処理ルールに従って、前記ユーザインタフェースにおける所定の設定項目の状態の直接変更の可否を制御する状態制御手段と、該状態制御手段により前記所定の設定項目の状態が直接変更不可とされた理由を含むメッセージを前記ユーザインタフェースを介して通知する通知制御手段と、を備え、前記コンフリクト処理ルールは、所定の設定項目の状態を直接変更不可とするための命令と、該命令によって前記直接変更不可とした理由を示す理由情報と、を含むことを特徴とする。

【0012】

【発明の実施の形態】（ハードウェア構成）図1は本発明の一実施形態を示す印刷処理システムのブロック構成図である。印刷処理システムは、ホストコンピュータ3

- 30 000と、プリンタ1500より構成される。

【0013】ホストコンピュータ3000において、1はシステムバス4に接続される各デバイスを総括的に制御するCPU、2はCPU1の主メモリ、ワークエリア等として機能するRAMである。3は各種プログラム、データを格納するROMであって、各種フォントを記憶するフォントROM3a、ブートプログラムやBIOS等を記憶するプログラムROM3b、および各種データを記憶するデータROM3cに区分けして構成されている。

- 40 【0014】5はキーボードコントローラ（KBC）で、キーボード9や不図示のポインティングデバイス（マウス）からのキー入力を制御する。6はCRTコントローラ（CRTC）であり、CRTディスプレイ（CRT）10の表示を制御する。

【0015】外部メモリ11（ディスクコントローラ（DKC）7によりアクセス制御される）は、ハードディスク（HD）やフロッピー（登録商標）ディスク（FD）等であり、図示の如く、オペレーティングシステムプログラム（以下OS）205をはじめ各種アプリケーション（例えば、図形、イメージ、文字、表等が混在し

た文書処理を行う文書処理アプリケーションプログラム) 201、印刷処理関連プログラム204を記憶する他、ユーザファイル、編集ファイル等も記憶する。印刷処理関連プログラム204は、ページ記述言語を用いて記述される印刷データを生成するプログラムであって、同系列の複数のプリンタに対して共通に利用されうる。また、プリンタ制御コマンド生成モジュール(以下「プリンタドライバ」という)2041、プリンタドライバUI制御モジュール2042を含む。

【0016】8はプリンタコントローラ(PTRC)であり、双方向性インターフェース21を介してプリンタ1500に接続され、プリンタ1500との通信制御処理を行う。

【0017】外部メモリ11に記憶されたアプリケーションは、RAM2にロードされてCPU1により実行されることになる。また、CPU1は、例えばRAM2へのアウトラインフォントの展開(ラスタライズ)処理を実行し、CRT10上のWYSIWYG(What you see is What you get)を可能としている。さらに、CPU1は、CRT10上の不図示のマウスカーソル等で指示されたコマンドに基づいて登録された種々のウインドウを開き、種々のデータ処理を実行する。ユーザは印刷を実行する際、印刷設定設定画面(プリンタドライバUI制御モジュール2042により制御される)を開き、プリンタの設定や、印刷モードの選択を含むプリンタドライバ2041に対する印刷処理の設定を行うことができる。

【0018】プリンタ1500において、12はプリンタ1500の全体を制御するCPUである。19はCPU12の主メモリ、ワークエリア等として機能とともに、出力情報展開領域、環境データ格納領域、NVRAM等に用いられるRAMであり、図示しない増設ポートに接続されるオプションRAMによりメモリ容量を拡張することができるよう構成されている。13はROMであり、各種フォントを記憶するフォントROM13a、制御プログラム等を記憶するプログラムROM13b、および各種データを記憶するデータROM13cにより構成される。

【0019】外部メモリ14(メモリコントローラ(MC)20によりアクセスを制御される)は、オプションとして接続されるハードディスク(HD)、フロッピーディスク(FD)、ICカード等であり、フォントデータ、エミュレーションプログラム、フォームデータ等を記憶する。ハードディスク等の外部メモリ14が接続されていない場合には、ROM13のデータROM13cに、ホストコンピュータ3000で利用される情報等を記憶することになる。なお、外部メモリ14は1個に限らず、複数備えるものであってもよく、例えば、内蔵フォントに加えてオプションフォントカード、言語系の異なるプリンタ制御言語を解釈するプログラム等を格納し

た外部メモリを複数接続できるように構成されていてよい。

【0020】操作部1501にはユーザからの操作を受け付ける操作パネルが設けられ、その操作パネルには操作のためのスイッチおよびLED表示器等が配されている(図示は省略)。また、図示しないNVRAMを有し、操作パネル1501からのプリンタモード設定情報を記憶するようにしてもよい。

【0021】プリンタCPU12は、ROM13のプログラムROM13bに記憶された制御プログラム等に基づき、印刷部インターフェース16を介してシステムバス15に接続される印刷部(プリンタエンジン)17に出力情報としての画像信号を出力する。また、CPU12は入力部18を介してホストコンピュータ3000との通信処理が可能となっており、プリンタ1500内の情報等をホストコンピュータ3000に通知可能に構成されている。

【0022】(ソフトウェア構成)図2は、所定のアプリケーションおよびプリンタ1500を制御対象とする

10印刷処理関連プログラムを起動して、ホストコンピュータ3000上のRAM2にロードされた状態のRAM2のメモリマップを示している。RAM2には、図示の如く、BIOS206、OS205をはじめ、アプリケーション201、印刷処理関連プログラム204、および関連データ203がロードされており、さらに空きメモリ領域202も確保されている。これにより、アプリケーション201および印刷処理関連プログラム204は実行可能状態にある。

【0023】印刷処理関連プログラム204におけるプリンタドライバUI制御モジュール2042は、ユーザによる印刷設定指令に応じてCRT10に印刷設定画面を表示しユーザからの設定を可能にする。

【0024】図8に、UIとしての印刷設定画面の表示例を示す。同図において、[Print Style]欄80は、印刷レイアウトを指定する欄であり、ユーザは例えば、片面印刷(1-Sided Printing)801、両面印刷(2-Sided Printing)802、および製本印刷(Booklet Printing)803のコントロールうちの、いずれかに対応するラジオボタンをマウスでクリックすることで指定することができる。

【0025】[Finishing]欄81は、印刷済み記録媒体の出力順序および仕上げについて指定する欄であり、ユーザは例えば、以下のいずれかから指定可能となっている。

【0026】[Collate]811
部単位印刷。Nページの文書をM部印刷する場合に、第1ページ、第2ページ、…、第Nページの順に1枚ずつ出力し、これをM回繰り返す。

【0027】[Group]812
ページ単位印刷。Nページの文書をM部印刷する場合

に、第1ページをM枚、第2ページをM枚、…、第NページをM枚、の順に出力する。

【0028】[Staple] 813

ステープル仕上げ。上記[Collate] 811と同様に部単位で出力し、仕上げとして各部ごとにステープラで止める。

【0029】なお、本明細書では、上記したようなユーザ設定可能項目を「プリンタ機能」または単に「機能」ともよぶ。この他にも多くのプリンタ機能を有するが、説明を簡単にするため省略する。

【0030】ここで、ユーザにとって不都合と思われる設定の組み合わせ、意味のない設定の組み合わせ、すなわち設定値間の不整合（コンフリクト）は、プリンタドライバUI制御モジュール2042により、以下詳細に説明するコンフリクト処理として回避されるように設計される。

【0031】図3は、実施形態における印刷処理関連プログラム204のプリンタドライバUI制御モジュール2042の概略構成を示している。303は、各モジュール間のデータの受け渡しやデータの更新等を管理してコンフリクト処理を統括するコンフリクトマネージャである。306が、上記した印刷設定画面表示制御を行い、本プリンタドライバUI制御モジュール2042におけるメインプログラムとしてのプリンタドライバUIである。301は、後述する記述形式で記述される不整合回避戦略を示すコンフリクト処理ルールを列記したコンフリクト処理ルール記述ファイルである。302は、コンフリクト処理ルール記述ファイル301をロードして、入力された設定値に対してコンフリクト処理ルールを適用し、各機能の状態を推論する推論エンジン、304は、各プリンタ機能の状態をリスト形式で表示する状態変数リストであり、ユーザからの入力およびコンフリクト処理ルール記述ファイル301の内容に基づき更新されうる。305は、プリンタドライバUI306が提供する画面表示の基になる帳票としての内部構造体であり、状態変数リスト304と連動して各プリンタ機能の状態を所定の形式で表示する。

【0032】その後に、プリンタドライバUI306を介してユーザからの設定情報を受け取ったコンフリクトマネージャ303は、コンフリクト処理ルール記述ファイル301を参照する。このことは、図示の如くコンフリクト処理ルール記述ファイル301からコンフリクトマネージャ303に向かう矢印で、「R(Read)」として表示されている。参照の結果、設定情報がコンフリクト処理ルールに適合する場合、コンフリクト処理が適用される。そして、コンフリクトマネージャ303は、状態変数リスト304および内部構造体305を更新してプリンタドライバUI306に反映させる。この更新作業は、図示の如くコンフリクトマネージャ303は、状態変数リスト304および内部構造体305とは各々双

方向矢印で結ばれ、「R/W (Read/Write)」として表示されている。

【0033】図4は、図3に示した各モジュールで扱われるデータの関連を説明する模式図である。図4において、コンフリクト処理ルール記述ファイル301は、推論エンジン302にインクルード（ロード）されたかたちで参照される。このコンフリクト処理ルール記述ファイル301はコンフリクトマネージャ303にも参照され、状態変数リスト304はコンフリクト処理ルールに

従った内容を有することになる。また、内部構造体305と状態変数リスト304とは先に述べたとおり連動表示されるものであるから、互いにマッピングされる関係にある。そしてこの状態がプリンタドライバUI306の制御によってユーザに見えるかたちで表現される。

【0034】状態変数リスト304について具体的に説明すると、同図に示すように、各プリンタ機能（同図中「キー」の欄A, B, C,...）毎に、状態値をはじめ、例えば、対応するコントロールの利用可否情報、利用不可の場合の理由（無効理由）情報、UI更新の要否情報等が記述される。

【0035】内部構造体305は、各プリンタ機能の状態を例えば、

データ型 機能名メンバ 状態メンバ

の構文で表し、プリンタ機能名A, B, Cに対応する機能名メンバをそれぞれ cA, cB, cC のように表現する。そして、各機能に対する状態メンバは、状態変数リスト304における状態値がマッピングされている。

【0036】（コンフリクト処理ルールの記述形式）次に、コンフリクト処理ルール記述ファイル301について説明する。

【0037】各ルールは、論理（ロジック）を用いて数学的に形式化される。述語は、

プリンタ機能名（引数）

の形で記述される。引数としては、ON/OFFの他、数値が入る場合もある（例えば、印刷部数等）。左辺には「プリンタ機能名（引数）」を、右辺には左辺が成り立つための論理条件を記述し、記号「←」で関係づける。例えば、

A(ON) ← B(ON).

は、「プリンタ機能Bの状態がONのときは、プリンタ機能Aの状態をONとする」という意味のルールになる。

【0038】また、式中の記号「,」は論理「かつ（AN D）」の意味で用いられる。例えば、「プリンタ機能Bの状態がON、かつ、プリンタ機能Cの状態がOFFのときに、プリンタ機能Aの状態をONとする」というルールは、A(ON) ← B(ON), C(OFF).

と記述される。

【0039】さらに、コンフリクト処理ルール記述ファイル301の中には、プリンタドライバUI306を更新するための処理を記述することが可能である。推論工

ンジン302がその記述を解釈した時点で、コンフリクトマネージャ303の状態変数リスト304を介して、プリンタドライバUI306の更新処理を直接行うこと が可能になっている。

【0040】 例えば、UI更新の処理として{disable} セットの記述を追加することにより、該当項目のコントロールを disableする処理（グレイアウト表示とともに設定不可の状態にする処理）を指示することができる。

【0041】 また、{disable} セットの後に{Reason(X)} セットの記述を追加すると、 disableとなった理由を表示するための理由識別子Xに応じたメッセージを表示させることができる（詳細は後述する）。

【0042】 次に、図8に表示されたプリンタ機能を例に、コンフリクト処理ルールに記述される機能名の具体例を示す。図8に示した[Collate] 811に対応する部単位印刷機能、[Group] 812に対応するページ単位印刷機能、および[Staple] 813に対応するステープル仕上げ機能の機能名はそれぞれ、 Collate()、 Group()、 およびStaple()で示され、引数はONまたはOFFとなる。また、[PrintStyle]欄80に対応する印刷レイアウト機能は、 Layout()で示され、引数は、1-Sided、2-Sided、 Bookletのいずれかである。

【0043】 この他、UIとは直接関連しない機能（例えば、特定のオプションの装着状態、セットされている記録用紙の種類等の属性を示す環境情報）を定義することができる。例えば、上記ステープル仕上げ機能に関連して、デバイスオプションとしてのステープル用フィニッシャが装着されているか否かを示すFinisher()を使用できる。引数はONまたはOFFであり、 ONのときはフィニッシャが装着されている状態を示し、 OFFのときは未装着の状態を示す。

【0044】 また、セットされた記録用紙がステープル可能な紙であるか否かを示すStapleablePaper()が使用可能である。引数はONまたはOFFであり、 ONのときはステープル可能な用紙がセットされている状態を示し、 OFFのときはステープル可能な用紙がセットされていない状態を示す。

【0045】 なお、上記した機能名はあくまでプリンタという制御対象に対する例であって、制御対象に応じてその他の機能名が定義することができることはいうまでもない。

【0046】 図6は、以上の例に従って記述されたコンフリクト処理ルール記述ファイル301の一例である。

【0047】 (1)は、フィニッシャが装着されていないときにはStaple(OFF)、すなわちステープル仕上げ不可、とするルールである。後続行には{disable}が記述されているので、このルールが適用されると、コントロール[Staple] 813はグレイアウトされることになる。また、次の行には、{Reason(NO_FINISHERR)}が記述され

ている。ここで、理由識別子NO_FINISHERRは、フィニッシャが装着されていないことを示すものである。

【0048】 (2)は、ステープル可能な記録用紙がセットされていないときはStaple(OFF)とするルールである。ここでも(1)と同様、後続行には{disable}が記述されているので、このルールが適用されると、コントロール[Staple] 813はグレイアウトされることになる。また、次の行には、ステープル不可能な用紙がセットされていることを示すINVALID_PAPERを理由識別子に有する

10 {Reason(INVALID_PAPER)} セットが記述されている。

【0049】 (3)は、[Booklet Printing] 803がチェックされたことでLayout(BOOKLET)となったときは、 Staple(OFF)とするルールである。ここでも同様に、このルールが適用されると、{disable}の記述によってコントロール[Staple] 813はグレイアウトされることになる。また、次の行には、適合しないプリントスタイル（レイアウト）が指定されていることを示すINVALID_LAYOUTを理由識別子に有する {Reason(INVALID_LAYOUT)} セットが記述されている。

20 【0050】 (4)は、記録媒体としてOHPシートがセットされたときは、 Staple(OFF)とするルールである。このルールが適用されると、{disable}の記述によってコントロール[Staple] 813はグレイアウトされる。また、次の行には、適合しない用紙タイプが指定されていることを示すINVALID_PAPERTYPEを理由識別子に有する {Reason(INVALID_PAPERTYPE)} セットが記述されている。

【0051】 (5)は、[Collate] 811がチェックされたことでCollate(ON)となったときにStaple(OFF)とする

30 ルール、(6)は、[Group] 812がチェックされたことでGroup(ON)となったときにStaple(OFF)とするルールである。

【0052】 (7)は、[Booklet Printing] 803がチェックされたことでLayout(BOOKLET)となったときにCollate(ON)とするルールであり、このルールが適用されると、次の行の{disable}の記述によってコントロール[Collate] 811がグレイアウトされる。

40 【0053】 (8)は、[Booklet Printing] 803がチェックされたことでLayout(BOOKLET)となったときにGroup(ON)とするルールであり、このルールが適用されると、次の行の{disable}の記述によってコントロール[Group] 812がグレイアウトされる。

【0054】 このように、各機能の状態によって適用されるルールが記述されている。

【0055】 (プリンタドライバUI制御モジュール2042の処理の内容)以下、図5のフローチャートを用いて、コンフリクト処理を含むプリンタドライバUI制御モジュール2042の処理について、詳しく説明する。

50 【0056】 プリンタドライバUI制御モジュール20

42の処理は、ユーザがキーボードコントローラKBC5等を用いて、印刷設定画面を開く指示をすることで始まる。ユーザが印刷設定画面を開くよう指示すると、上記したとおり、OS205の管理の下、RAM2に印刷処理関連プログラム204がロードされる。

【0057】印刷処理関連プログラム204がRAM2にロードされると、まず、印刷設定画面を開くための初期化処理として、推論エンジン302は、コンフリクト処理ルール記述ファイル301をコンフリクトマネージャ303を介して、RAM2に読み込む（ステップS501）。

【0058】続いて、コンフリクトマネージャ303が利用する状態変数リスト304を作成する（ステップS503）。

【0059】コンフリクト処理ルール記述ファイル301に記述されるすべてのプリンタ機能名は、コンフリクトマネージャ303の内部にインクルードされた状態変数リスト304にそれぞれ、状態変数を持っている。この状態変数の値は、プリンタドライバUI306で使用される内部構造体305の対応するメンバの値と連動している。各プリンタ機能名の状態変数の初期値は、その内部構造体305のメンバの値となる。

【0060】例えば、図4において、内部構造体305に記述されているint caの初期値は0であるから、それに対応する状態変数リスト304におけるプリンタ機能Aの値はOFFとなっている。したがって、状態変数リスト304におけるプリンタ機能Aの状態の初期値はOFFとなる。同様に、プリンタ機能名Bの初期値はON、プリンタ機能名Cの初期値はOFFとなる。すなわち、

A OFF

B ON

C OFF

となる。

【0061】その後、推論エンジン302は、コンフリクト処理ルール記述ファイル301を参照することでコンフリクトチェックの推論を行う。例えば、図4に示すように、コンフリクト処理ルール記述ファイル301に記述されている、

A(ON) ← B(ON), C(OFF).

が成立した場合、推論エンジン302は、状態変数リスト304中のプリンタ機能Aの状態変数値を、初期値OFFからONに変更する。すなわち、

A ON

B ON

C OFF

となる。このように状態変数リスト304が初期化される。また、状態変数リスト304中の他の項目（コントロールの表示／非表示情報等）も同様に初期化される。

【0062】コンフリクトチェックの推論が終了した後、コンフリクトマネージャ303は、変更された状態

変数の値を内部構造体305の対応するメンバ int caに反映させる。すなわち、int caは上記ルールが成立したことによって0から1に変更されることになる。

【0063】具体的に、図8に示したUIを開く際の初期化処理を例にとって説明する。図7は、図8のUIに対する内部構造体の一例を示している。同図から分かるように、例えば機能名メンバLayout、Collate、Group、およびStapleに対応する状態メンバは各々、1Sided、0、1、および0となっている。この状態から、図6に

10 示したコンフリクト処理ルール記述ファイル301を参照してコンフリクトチェックを行う。内部構造体の内容が図7に示す状態である場合には、図6に示したコンフリクト処理ルール記述ファイル301のうち、いずれのルールも適用される場合はない。つまり、上記機能名の状態変数値は、ステップS503の初期化による初期値として、次のようになる。

Collate OFF

Group ON

Staple OFF

20 Layout 1-Sided

【0064】続いて、UIのオープンのために必要とされるその他の初期化処理を行い、図8に例示したようなUIをオープンする（ステップS504）。この場合、状態変数リスト304を参照して、設定値に応じた値を表示したり、コントロールの利用可否情報に応じた処理（able/disableの変更）等を行うことになる。

【0065】UIとしての印刷設定画面がオープンされた後は、OSより送られてくるイベントを取得し、そのイベントに対する処理を行う（ステップS505）。

30 【0066】次に、ステップS505で取得したイベントが、ユーザが印刷設定画面上の設定項目を変更したイベントであるか否かの判別を行う（ステップS506）。ここで、取得したイベントがユーザによる設定変更要求であった場合には、ステップS507に進み、コンフリクト処理ルール記述ファイル301を参照してコンフリクト処理ルールを適用する。

【0067】取得したイベントがユーザによる設定変更要求であった場合の一例として、図8に示す[Print Style]欄80における、片面印刷（1-Sided Printing）8

40 01から製本印刷（Booklet Printing）803に変更するものであった場合について説明する。ここでは、ステープル仕上げが可能なフィニッシャが装着され、用紙サイズおよび用紙タイプがステープル可能なものがセットされているものとする。

【0068】プリンタ機能名Collate、Group、Staple、Layoutの状態変数の設定変更要求前における値は、図7に示した状態、すなわち、以下になっていた。

Collate OFF

Group ON

50 Staple OFF

Layout 1-Sided

【0069】ユーザの変更要求がLayoutを1-SidedからBookletに変更するものであるので、Layoutの状態変数の内容が変更されて、各プリンタ機能名の状態変数の値は次のようになる。

Collate OFF

Group ON

Staple OFF

Layout Booklet

【0070】すると、プリンタドライバUI306はコンフリクトマネージャ303をコールし、続いて推論エンジン302がコールされて、コンフリクト処理ルール記述ファイル301を参照してLayoutを起点としたコンフリクトチェックが始まる。

【0071】推論エンジン302は、コンフリクト処理ルール記述ファイル301の各ルールの適用／不適用をチェックしていく。上記したとおり、この例ではステープル仕上げが可能なフィニッシャが装着され、用紙サイズおよび用紙タイプがステープル可能なものがセットされているので、ルール(1)、(2)が適用されることはない。

【0072】ルール(3)は適用されることになるから、StapleはOFFのままコントロールはdisable、すなわちグレイアウト表示によって設定不可の状態にされる。さらに、上記したように、グレイアウトされた理由として不適合なプリントスタイルが指定されていることを示すINVALID_LAYOUTがセットされる。

【0073】ルール(4)、(5)、(6)は適用されないが、ルール(7)、(8)が適用され、Collateの値はOFFからONへ、Groupの値はONからOFFへ変更され、次のように各コントロールはdisableとなる。

Collate ON (disable)

Group OFF (disable)

Staple OFF (disable, Reason=INVALID_LAYOUT)

Layout BOOKLET

【0074】また、同じコントロールをグレイアウトするものではあるがその理由が相違するルールが、重複して適用される場合もある。この場合、あらかじめ定めた優先順位が高い理由が採用される。実施形態では、コンフリクト処理ルール記述ファイル301に先に記述されたルールほど優先度が高いと定めておく。

【0075】例えば、上記した例に統いて、ステープル仕上げ不可能なサイズの用紙がセットされた場合を考える。この場合、ルール(3)とともに、ルール(2)が適用されることになる。実施形態によれば、優先順位の高いルール(2)を採用して、グレイアウトされた理由としては、不適合な記録用紙がセットされていることを示すINVALID_PAPERがセットされることになる。すなわち、各コントロールの状態は、次のようになる。

Collate ON (disable)

Group OFF (disable)

Staple OFF (disable, Reason=INVALID_PAPER)

Layout BOOKLET

【0076】以上で、ステップS507の、推論エンジン302でのコンフリクト処理ルールの適用が終了する。

【0077】コンフリクトマネージャ303は、上記ステップS507のコンフリクト処理ルールの適用結果に基づき状態変数リスト304を更新する（ステップS508）。すなわち、状態変数リストは図11に示すようになる。図11において、機能Stapleの状態がOFF、そのコントロールがdisableとなるとともに、その理由識別子INVALID_PAPERが記録されていることが分かる。

【0078】続いて、内部構造体305を更新する（ステップS509）。具体的には、状態変数リスト304中のUI更新情報が「要(yes)」となっているコントロールについてはUIの更新処理を行い、その後UI更新情報は「不要(no)」に変更される。上記の例では、Layoutが1-Sided PrintingからBooklet Printingに設定が変更されたことで、CollateがOFFからONへ、GroupがONからOFFへと変更され、Collate、Group、Staple がそれぞれdisableとなり、図8のUIは図9に示すとおりに更新される。

【0079】また、図9に示すように、コントロールがグレイアウトされた場合には、そのことを知らせるためのコンフリクトマーク91が表示される（詳細は後述する）。ステップS509の後はステップS505に戻って処理を繰り返す。

【0080】ステップS506において、ステップS505で取得したイベントがユーザが印刷設定画面上の設定項目を変更したイベントではなかったときは、ステップS512に進み、グレイアウトの理由表示のイベントであるかを判別する。このイベントは、例えばコンフリクトマーク91がクリックされたときやコンフリクトマーク91上にマウスカーソルが重なったときなどに発生する。

【0081】このイベントが発生した場合は、ステップS513に進み、プリンタドライバUI306は状態変数リスト304から該当するコントロールの無効理由情報を参照し、そこに登録されている識別子に対応する処理を行う。例えば、図10に示すようなコメントメッセージ92を表示することができる。いずれかのキーを押下したりマウスカーソルを移動するなどによりイベントが解除されると、メッセージは消え、ステップS505に戻って処理を繰り返す。

【0082】ステップS512の判別の結果、理由表示イベントではなかった場合には、ステップS514に進み、プリンタドライバUIのクローズ要求かどうかの判別を行う。クローズ要求であった場合には、状態変数の設定で内部構造体を更新し（ステップS515）、所定

の終了処理を行い、プリンタドライバUIをクローズして、すべての処理を終了する（ステップS516）。一方、クローズ要求ではなかった場合には、ステップS505に戻って処理を繰り返す。

【0083】以上の処理は、プリンタドライバUIがクローズされるまで、繰り返し実行される。プリンタドライバUIがクローズされると処理は終了し、印刷処理関連プログラム204の処理も終了し、RAM2からはOS205の機能により消去される。

【0084】以上説明した実施形態によれば、コンフリクト処理ルール記述ファイルに、コンフリクト処理ルールの適用によってコントロールをdisableして当該コントロールをグレイアウトする場合にはその理由を記述することを可能にしたので、グレイアウトされた理由情報の管理を容易にできるようになった。このため、理由検知のための専用プログラムコードを追加する煩わしさや、条件の漏れや食い違いを生ずるケースから解放され、品質の高い理由表示処理を実現することができる。

【0085】さらに他の発明によれば、理由表示処理をプリンタドライバUIからコンフリクトマネージャに移することで両者の間で理由識別子の同期を取る必要がなくなり、その結果独立性が向上し、生産性が高くメンテナンス性に優れるという効果がある。

【0086】

【他の実施形態】上述した実施形態では、ステップS513の理由コメントの表示はプリンタドライバUI306が行うようにしたが、コンフリクトマネージャ303が行うようにすることもできる。すなわち、コンフリクトマネージャ303の内部に文字列リソースを持ち、理由表示イベント発生時にはコンフリクトマネージャ303を呼び出し、そこで状態変数の理由属性を参照して該当するメッセージを表示するようにしてもよい。

【0087】また、ステップS507では、最も優先順位の高い理由をひとつだけ保持しているが、該当する理由をすべて保持し、ステップS513にてすべて表示するようにしてもよい。

【0088】また、ステップS507では、ルールの記述順序によって優先順位を定めるようにしたが、もちろんこれに限定されるものではなく、別の基準を設けてもよい。状態値を変更するルールのみを適用することで、最新の理由を保持すると共に効率化を図ることもできる。

【0089】なお、上述した実施形態では、プリンタ装置に対して、コンフリクト処理を含むUI制御を行う例について説明したが、本発明は、プリンタ装置に限らず、デジタルカメラ、デジタルレコーダ、イメージスキャナ等の周辺装置、制御機器の他、モ뎀やルータといったネットワーク関連機器にも適用可能であることはいうまでもない。これらの複数の機器から構成される

システムに適用することも可能である。

【0090】上述したように、本発明の目的は、前述した実施形態の機能を実現するソフトウェアのプログラムコードを記録した記憶媒体（または記録媒体）を、システムあるいは装置に供給し、そのシステムあるいは装置のコンピュータ（またはCPUやMPU）が記憶媒体に格納されたプログラムコードを読み出し実行することによっても、達成される。この場合、記憶媒体から読み出されたプログラムコード自体が前述した実施形態の機能

10 を実現することになり、そのプログラムコードを記憶した記憶媒体は本発明を構成することになる。また、コンピュータが読み出したプログラムコードを実行することにより、前述した実施形態の機能が実現されるだけでなく、そのプログラムコードの指示に基づき、コンピュータ上で稼働しているオペレーティングシステム(OS)などが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0091】さらに、記憶媒体から読み出されたプログラムコードが、コンピュータに挿入された機能拡張カードやコンピュータに接続された機能拡張ユニットに備わるメモリに書込まれた後、そのプログラムコードの指示に基づき、その機能拡張カードや機能拡張ユニットに備わるCPUなどが実際の処理の一部または全部を行い、その処理によって前述した実施形態の機能が実現される場合も含まれることは言うまでもない。

【0092】本発明を上記記憶媒体に適用する場合、その記憶媒体には、先に説明した図5のフローチャートに対応するプログラムコードが格納されることになる。

【0093】

【発明の効果】以上説明したように、ユーザインタフェースにおいて特定の項目が設定不可となった理由を表示する機能を、理由検知の漏れや食い違いを起こすことなく実現し、開発者の作業工数や人為的ミスを低減することの可能なユーザインタフェース制御装置および方法を提供することができる。

【図面の簡単な説明】

【図1】実施形態に係る印刷処理システムのブロック構成図である。

40 【図2】実施形態におけるRAM2のメモリマップを示す図である。

【図3】実施形態におけるプリンタドライバUI制御モジュールの概略構成図である。

【図4】実施形態におけるプリンタドライバUI制御モジュールで扱われるデータの関連を説明する模式図である。

【図5】実施形態におけるプリンタドライバUI制御モジュールの処理を示すフローチャートである。

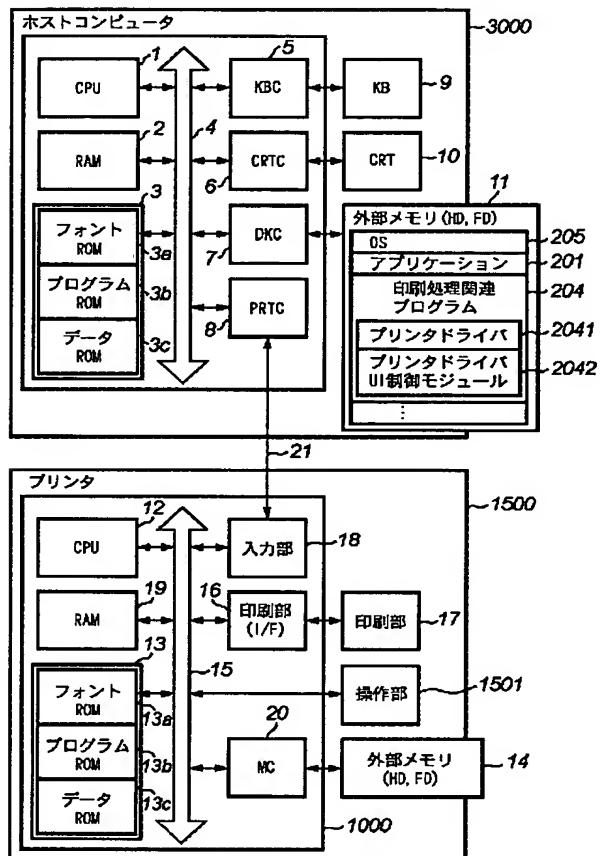
【図6】実施形態におけるコンフリクト処理ルール記述ファイルの一例を示す図である。

【図7】実施形態における内部構造体の一例を示す図である。

【図8】実施形態におけるプリンタドライバUIによる印刷設定画面の一例を示す図である。

【図9】実施形態におけるプリンタドライバUIによる

【図1】



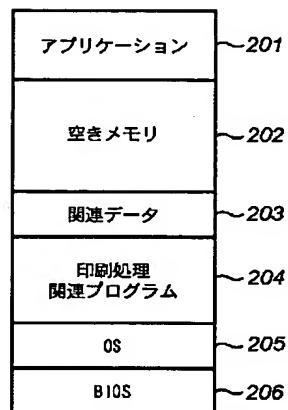
【図1】

印刷設定画面の一例を示す図である。

【図10】実施形態におけるプリンタドライバUIによる印刷設定画面の一例を示す図である。

【図11】実施形態における状態変数リストの一例を示す図である。

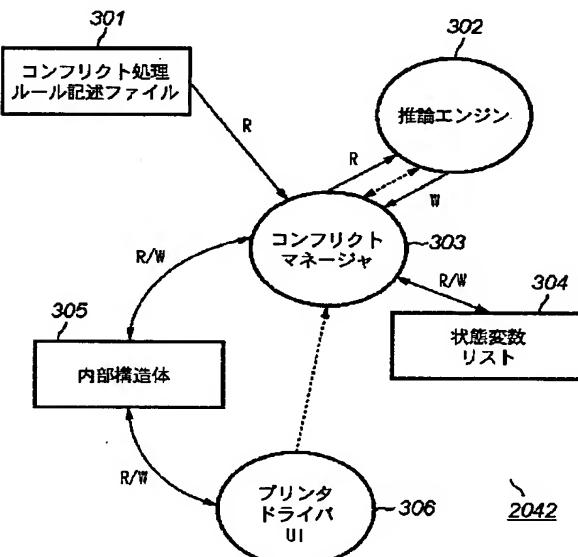
【図2】



【図7】

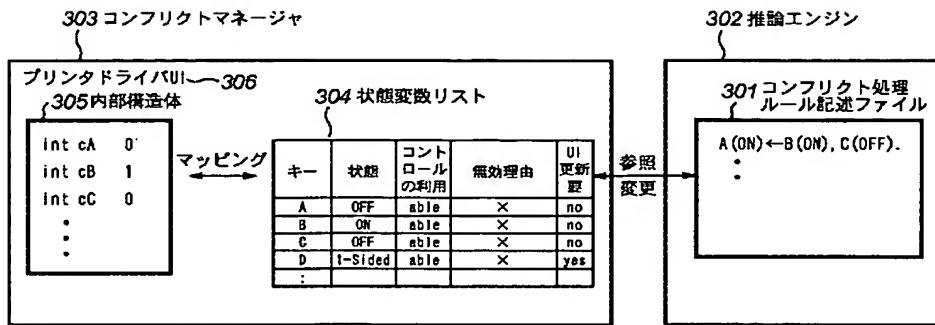
Int	cLayout	1Sided
Int	cCollate	0
Int	cGroup	1
Int	cStaple	0
BOOL	bFinisher	TRUE
:		

【図3】

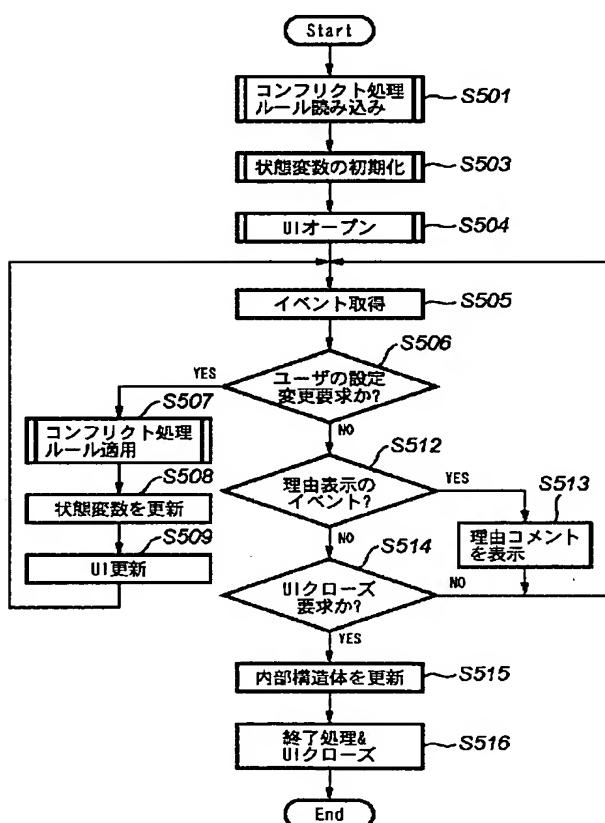


キー	状態	コントロールの利用	無効理由	UI更新要
Collate	ON	disable	×	yes
Group	OFF	disable	×	yes
Staple	OFF	disable	INVALID_PAPER	yes
Layout	BOOKLET	able		yes
:				

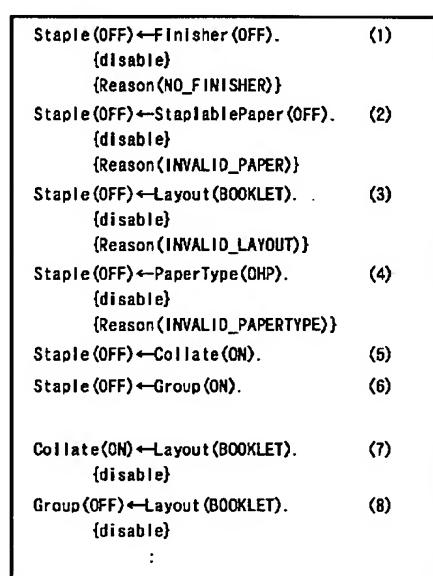
【図 4】



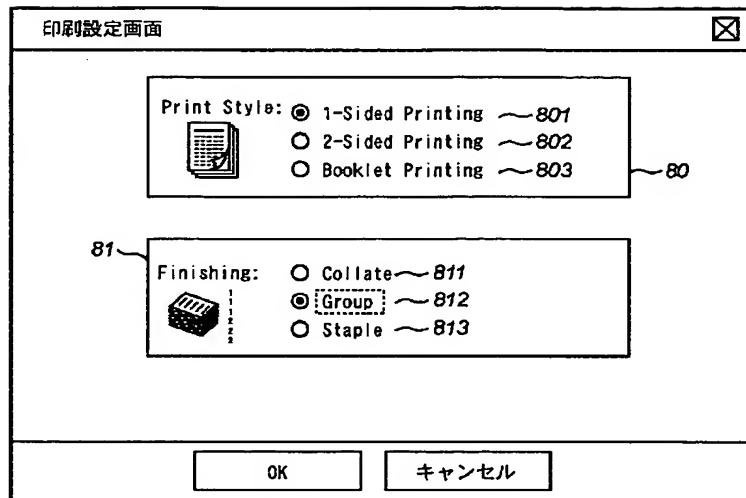
【図 5】



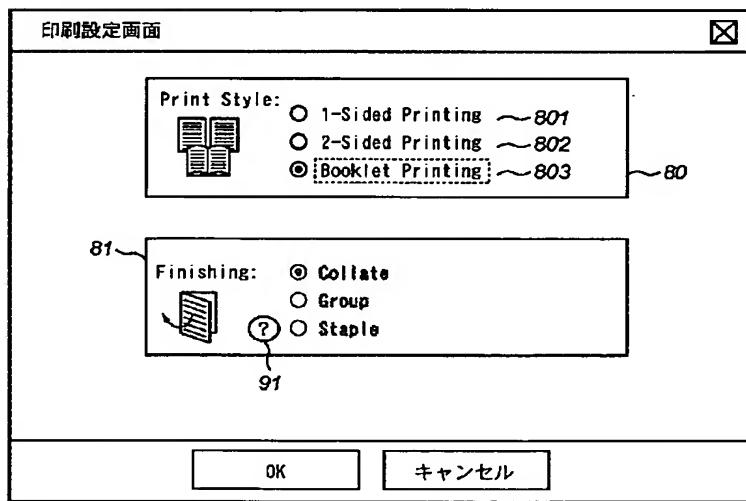
【図 6】



【図8】



【図9】



【図10】

